Remarks

Claims 1-22 are at issue. Claims 1-22 stand rejected under 35 USC 103(a) as being unpatentable over http://www.xml.com/pub/a/2000/01/10/perlwebtools.html (hereinafter Hampton).

Claim 1 requires a map index. A map index is shown if FIGs. 3 & 8 and is a series of pointers to a dictionary where the tags and data are stored. Hampton only discusses creating a hash directory of the files (HTML files) where the directory is the key and the list of files in that directory is the value. This is clearly not a flatten data structure (numerical DOM), since it does not suggest flattened the files or even indexing the files. It is a simple (hashed) index between the directory and the files. There is no discussion of a map index or a map at all. Nor does Hampton discuss a map offset. Clearly, claim 1 is allowable.

Claim 2 requires that the identified couplet of the numerical DOM be converted into an XML string. The application clearly defines a numerical DOM (page 10, lines 15-16) as a flattened structured data document that is stored as series of pointers and offsets (see FIG. 3 & 8). FIG. 2 of the application shows an example of a flattened data document. The flattened data document of FIG. 2 is an example of how to flatten the XML document shown in FIG. 1 of the present application. Hampton discusses converting your HTML documents into XML. He never discusses flattening the XML to form a numerical DOM. Hampton never uses the term numerical DOM. Hampton does discuss creating a hash directory of the files (HTML files) where the directory is the key and the list of files in that directory is the value. This is clearly not a flatten data structure (numerical DOM), since it does not suggest flattened the files or even indexing the files. It is a simple (hashed) index between the directory and the files. Claim 2 is allowable over the prior art.

Claim 3 requires converting the data couplet into a data XML string. There is no discussion in Hampton of a data couplet. A data couplet (FIG. 14, steps 358 & 360) is found using the map offset. The map 202 (FIG. 8) has pointers that point to the dictionary to find the data and tags. Since, Hampton does not discuss flattening the XML document he would not have a data couplet that needed to be converted back to an XML string. Claim 3 is allowable.

Claim 4 requires that when a query is partially qualified, transforming a target into a partially qualified hashing code. There is no discussion of partial or fully qualified search targets in Hampton and no discussion of partially qualified hashing codes. Claim 4 is allowable.

Claim 5 requires a wildcard target in the query. There is no discussion of how wildcards are treated in Hampton or any indication that they are allowed. Claim 5 is allowable.

Claim 6 requires a sliding window search on a dictionary when the target type is an incomplete data string. There is no discussion in Hampton of sliding window searches, dictionaries or incomplete data strings. Claim 6 is allowable over the prior art.

Claim 7 requires returning a plurality of dictionary offsets. Hampton never discusses a dictionary or dictionary offsets. Claim 7 is allowable.

Claim 8 requires a map index and a map offset. The map of the present application is discussed with respect to FIGs. 3 & 8. Hampton does not discuss a map, map index or map offset. Claim 8 is allowable.

Claim 9 requires a map index. The map of the present application is discussed with respect to FIGs. 3 & 8. Hampton does not discuss a map, map index or map offset. Claim 9 is allowable.

Claims 10 & 11 are allowable as being dependent upon an allowable base claim.

Claim 12 required creating a numerical DOM. The application clearly defines a numerical DOM (page 10, lines 15-16) as a flattened structured data document. FIG. 2 of the application shows an example of a flattened data document that is stored as series of pointers and offsets (see FIG. 3 & 8). The flattened data document of FIG. 2 is an example of how to flatten the XML document shown in FIG. 1 of the present application. Hampton discusses converting your HTML documents into XML. He never discusses flattening the XML to form a numerical DOM. Hampton never uses the term numerical DOM. Hampton does discuss creating a hash directory of the files (HTML files) where the directory is the key and the list of files in that directory is the value. This is clearly not a flatten data structure (numerical DOM), since it does not suggest flattened the files or even indexing the files. It is a simple (hashed) index between the directory and the files. Claim 12 is allowable over the prior art.

Claim 13 requires dictionary offsets and pointers. Hampton never discusses a dictionary or dictionary offsets or pointer. Claim 13 is allowable.

Claim 14 requires an alias request and a dictionary. Hampton never discusses a dictionary or an alias request. Claim 14 is allowable.

Claim 15 requires an alias index. Hampton never discusses an alias index. Claim 15 is allowable.

Claim 16 requires a dictionary. Hampton never discusses a dictionary. Claim 16 is allowable.

Claim 17 requires a dictionary. Hampton never discusses a dictionary. Claim 17 is allowable.

Claim 18 requires a map index and a map offset. The map of the present application is discussed with respect to FIGs. 3 & 8. Hampton does not discuss a map, map index or map offset. Claim 18 is allowable.

Claim 19 requires a numerical DOM. The application clearly defines a numerical DOM (page 10, lines 15-16) as a flattened structured data document. FIG. 2 of the application shows an example of a flattened data document that is stored as series of pointers and offsets (see FIG. 3 & 8). The flattened data document of FIG. 2 is an example of how to flatten the XML document shown in FIG. 1 of the present application. Hampton discusses converting your HTML documents into XML. He never discusses flattening the XML to form a numerical DOM. Hampton never uses the term numerical DOM. Hampton does discuss creating a hash directory of the files (HTML files) where the directory is the key and the list of files in that directory is the value. This is clearly not a flatten data structure (numerical DOM), since it does not suggest flattened the files or even indexing the files. It is a simple (hashed) index between the directory and the files. Claim 19 is allowable over the prior art.

Claim 20 requires a dictionary index and a dictionary offset. Hampton never discusses a dictionary or dictionary offsets or index. Claim 20 is allowable.

Claim 21 requires converting the data couplet into a data XML string. There is no discussion in Hampton of a data couplet. A data couplet (FIG. 14, steps 358 & 360) is found using the map offset. The map 202 (FIG. 8) has pointers that point to the dictionary to find the data and tags. Since, Hampton does not discuss flattening the XML document he would not have a data couplet that needed to be converted back to an XML string. Claim 21 is allowable.

Claim 22 requires a wildcard and numerical DOM. The application clearly defines a numerical DOM (page 10, lines 15-16) as a flattened structured data document that is stored as series of pointers and offsets (see FIG. 3 & 8). FIG. 2 of the application shows an example of a flattened data document. The flattened data document of FIG. 2 is an example of how to flatten the XML document shown in FIG. 1 of the present application. Hampton discusses converting your HTML documents into XML. He never discusses flattening the XML to form a numerical DOM. Hampton never uses the term numerical DOM. Hampton does discuss creating a hash directory of the files (HTML files) where the directory is the key and the list of files in that directory is the value. This is clearly not a flatten data structure (numerical DOM), since it does not suggest flattened the files or even indexing the files. It is a simple (hashed) index between the directory and the files. There is no discussion of how wildcards are treated in Hampton or any indication that they are allowed. Claim 22 is allowable over the prior art.

The application has been placed in condition for allowance, prompt reconsideration and allowance are respectfully requested.

Respectfully submitted,

(Brandin)

By ________________________
Dale B. Halling
Attorney for the Applicant
Registration No. 38,170
Phone: (719) 447-1990
Fax: (719) 447-9815

I hereby certify that a Response is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner of Patents and Trademarks, P.O. Box 1450, Alexandria, VA 22313-1450, on:

10/22/09
Date

Signature (Dale B. Halling)